ABSTRACT

An optimizing compiler and method thereof performs a sequence of optimizing changes to an intermediate language representation of a routine, and measures an execution characteristic of each optimization, such as a timing of the machine language representation performed on an architecture similar to the target machine using a user selectable initialization state. The sequence of optimizations is selected according to a criterion that includes a lexicographic search and other methods. The pre-optimized code is also broken into segments wherein discrete optimizations are performed on each segment and measured using a user provided routine. The target routine is tested with the object code in main memory if not the cache if possible and optimizations are chosen only if they improve the target subroutine according to the user defined metric. After a stopping criterion is achieved, the most optimized code is selected.